MEDICAL INVENTORY OPTIMIZATION

PREPROCESSING AND EDA USING SQL

** PREPROCESSING

1. Creating a database in SQL using MySQL. The query for creating a database is

   ```
   Create database medical_inventory;
   ```

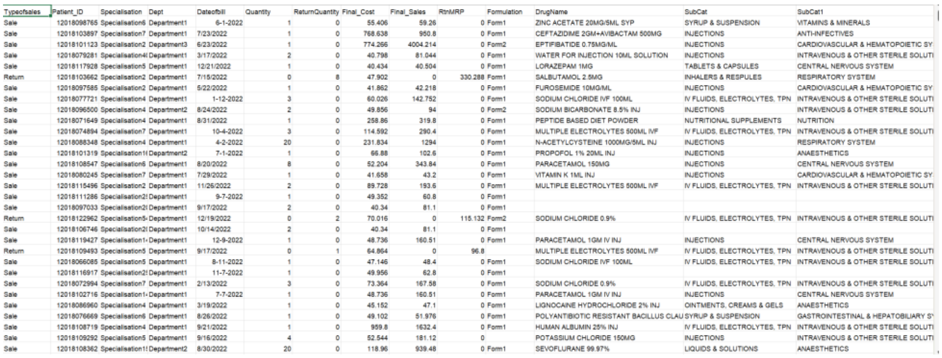
2. Importing data into the medical inventory database and naming the table as medical_data.

3. To use medical_data we need to run a query that uses medical inventory database

   ```
   use medical_inventory;
   ```

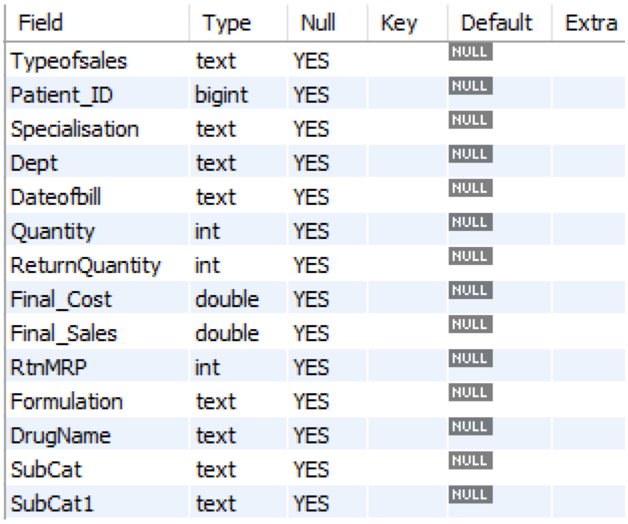
4. Taking a look at the data using select statement and finding the data types using describe statement.

   ```
   Select * from medical_data;
   ```



Typeofsales	Patient_ID	Specialisation	Dept	Dateofbill	Quantity	ReturnQuantity	Final_Cost	Final_Sales	RtnMRP	Formulation	DrugName	SubCat	SubCat1
Sale	12018098765	Specialisation6	Department1	6-1-2022	1	0	55.406	59.26	0	Form1	ZINC ACETATE 20MG/5ML SYP	SYRUP & SUSPENSION	VITAMINS & MINERALS
Sale	12018103897	Specialisation7	Department1	7/23/2022	1	0	768.638	950.8	0	Form1	CEFTAZIDIME 2GM+AVIBACTAM 500MG	INJECTIONS	ANTI-INFECTIVES
Sale	12018101123	Specialisation2	Department3	6/23/2022	1	0	774.266	4004.214	0	Form2	EPTIFIBATIDE 0.75MG/ML	INJECTIONS	CARDIOVASCULAR & HEMATOPOIETIC SY
Sale	12018079281	Specialisation4	Department1	3/17/2022	2	0	40.798	81.044	0	Form1	WATER FOR INJECTION 10ML SOLUTION	INJECTIONS	INTRAVENOUS & OTHER STERILE SOLUTI
Sale	12018117928	Specialisation5	Department1	12/21/2022	1	0	40.434	40.504	0	Form1	LORAZEPAM 1MG	TABLETS & CAPSULES	CENTRAL NERVOUS SYSTEM
Return	12018103662	Specialisation2	Department1	7/15/2022	0	8	47.902	0	330.288	Form1	SALBUTAMOL 2.5MG	INHALERS & RESPULES	RESPIRATORY SYSTEM
Sale	12018097585	Specialisation2	Department1	5/22/2022	1	0	41.862	42.218	0	Form1	FUROSEMIDE 10MG/ML	INJECTIONS	CARDIOVASCULAR & HEMATOPOIETIC SY
Sale	12018077721	Specialisation4	Department1	1-12-2022	3	0	60.026	142.752	0	Form1	SODIUM CHLORIDE IVF 100ML	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTI
Sale	12018096500	Specialisation4	Department2	8/24/2022	2	0	49.856	94	0	Form2	SODIUM BICARBONATE 8.5% INJ	INJECTIONS	INTRAVENOUS & OTHER STERILE SOLUTI
Sale	12018071649	Specialisation4	Department1	8/31/2022	1	0	258.86	319.8	0	Form1	PEPTIDE BASED DIET POWDER	NUTRITIONAL SUPPLEMENTS	NUTRITION
Sale	12018074894	Specialisation7	Department1	10-4-2022	3	0	114.592	290.4	0	Form1	MULTIPLE ELECTROLYTES 500ML IVF	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTI
Sale	12018088348	Specialisation4	Department1	4-2-2022	20	0	231.834	1294	0	Form1	N-ACETYLCYSTEINE 1000MG/5ML INJ	INJECTIONS	RESPIRATORY SYSTEM
Sale	12018101319	Specialisation1	Department2	7-1-2022	1	0	66.88	102.6	0	Form1	PROPOFOL 1% 20ML INJ	INJECTIONS	ANAESTHETICS
Sale	12018108547	Specialisation6	Department1	8/20/2022	8	0	52.204	343.84	0	Form1	PARACETAMOL 150MG	INJECTIONS	CENTRAL NERVOUS SYSTEM
Sale	12018080245	Specialisation7	Department1	7/29/2022	1	0	41.658	43.2	0	Form1	VITAMIN K 1ML INJ	INJECTIONS	CARDIOVASCULAR & HEMATOPOIETIC SY
Sale	12018115496	Specialisation2	Department1	11/26/2022	2	0	89.728	193.6	0	Form1	MULTIPLE ELECTROLYTES 500ML IVF	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTI
Sale	12018111286	Specialisation2	Department1	9-7-2022	1	0	49.352	60.8	0	Form1			
Sale	12018097033	Specialisation2	Department1	9/17/2022	2	0	40.34	81.1	0	Form1			
Return	12018122962	Specialisation5	Department1	12/19/2022	0	2	70.016	0	115.132	Form2	SODIUM CHLORIDE 0.9%	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTI
Sale	12018106746	Specialisation2	Department1	10/14/2022	2	0	40.34	81.1	0	Form1			
Sale	12018119427	Specialisation1	Department1	12-9-2022	1	0	48.736	160.51	0	Form1	PARACETAMOL 1GM IV INJ	INJECTIONS	CENTRAL NERVOUS SYSTEM
Return	12018109493	Specialisation5	Department1	9/17/2022	0	1	64.864	0	96.8		MULTIPLE ELECTROLYTES 500ML IVF	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTI
Sale	12018066085	Specialisation5	Department1	8-11-2022	1	0	47.146	48.4	0	Form1	SODIUM CHLORIDE IVF 100ML	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTI
Sale	12018116917	Specialisation2	Department1	11-7-2022	1	0	49.956	62.8	0	Form1			
Sale	12018072994	Specialisation7	Department1	2/13/2022	3	0	73.364	167.58	0	Form1	SODIUM CHLORIDE 0.9%	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTI
Sale	12018102716	Specialisation1	Department1	7-7-2022	1	0	48.736	160.51	0	Form1	PARACETAMOL 1GM IV INJ	INJECTIONS	CENTRAL NERVOUS SYSTEM
Sale	12018086960	Specialisation4	Department1	3/19/2022	1	0	45.152	47.1	0	Form1	LIGNOCAINE HYDROCHLORIDE 2% INJ	OINTMENTS, CREAMS & GELS	ANAESTHETICS
Sale	12018076669	Specialisation6	Department1	8/26/2022	1	0	49.102	51.976	0	Form1	POLYANTIBIOTIC RESISTANT BACILLUS CLAU	SYRUP & SUSPENSION	GASTROINTESTINAL & HEPATOBILIARY SY
Sale	12018108719	Specialisation4	Department1	9/21/2022	1	0	959.8	1632.4	0	Form1	HUMAN ALBUMIN 25% INJ	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTI
Sale	12018109292	Specialisation5	Department1	9/16/2022	4	0	52.544	181.12	0		POTASSIUM CHLORIDE 150MG	INJECTIONS	INTRAVENOUS & OTHER STERILE SOLUTI
Sale	12018108362	Specialisation1	Department2	8/30/2022	20	0	118.96	939.48	0	Form1	SEVOFLURANE 99.97%	LIQUIDS & SOLUTIONS	ANAESTHETICS

```
Describe medical_data;
```



Field	Type	Null	Key	Default	Extra
Typeofsales	text	YES		NULL	
Patient_ID	bigint	YES		NULL	
Specialisation	text	YES		NULL	
Dept	text	YES		NULL	
Dateofbill	text	YES		NULL	
Quantity	int	YES		NULL	
ReturnQuantity	int	YES		NULL	
Final_Cost	double	YES		NULL	
Final_Sales	double	YES		NULL	
RtnMRP	int	YES		NULL	
Formulation	text	YES		NULL	
DrugName	text	YES		NULL	
SubCat	text	YES		NULL	
SubCat1	text	YES		NULL	

5. Renaming the ï»¿Typeofsales to Typeofsales column.

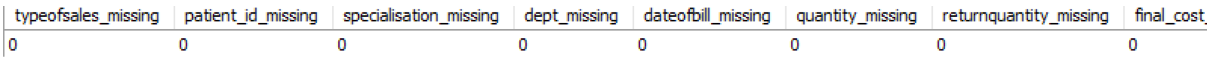
```
alter table medical_data rename column ï»¿Typeofsales to Typeofsales;
```

6. Creating a new table with name clean_data and doing all the preprocessing and storing in the clean_data table. Dateofbill column is in text format we need it in date format.

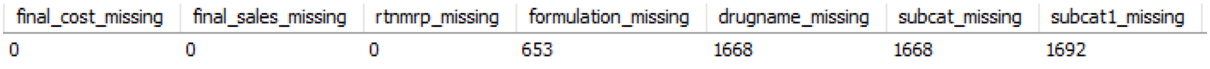
```
CREATE TABLE clean_data AS
SELECT Typeofsales, Patient_ID, Specialisation, Dept,
 STR_TO_DATE(REPLACE(Dateofbill,'/','-'),'%m-%d-%Y') AS Dateofbill,
Quantity, ReturnQuantity,
 Final_Cost, Final_Sales, RtnMRP, Formulation, DrugName, SubCat,
SubCat1
FROM medical_data;
```

7. Checking for null values

```
SELECT
 COUNT(CASE WHEN TRIM(Typeofsales) = '' OR Typeofsales IS NULL THEN
1 END) AS
typeofsales_missing,
 COUNT(CASE WHEN Patient_ID IS NULL THEN 1 END) AS
patient_id_missing,
 COUNT(CASE WHEN TRIM(Specialisation) = '' OR Specialisation IS NULL
THEN 1 END) AS
specialisation_missing,
 COUNT(CASE WHEN TRIM(Dept) = '' OR Dept IS NULL THEN 1 END) AS
dept_missing,
 COUNT(CASE WHEN TRIM(Dateofbill) = '' OR Dateofbill IS NULL THEN 1
END) AS dateofbill_missing,
 COUNT(CASE WHEN Quantity IS NULL THEN 1 END) AS quantity_missing,
 COUNT(CASE WHEN ReturnQuantity IS NULL THEN 1 END) AS
returnquantity_missing,
 COUNT(CASE WHEN Final_Cost IS NULL THEN 1 END) AS
final_cost_missing,
 COUNT(CASE WHEN Final_Sales IS NULL THEN 1 END) AS
final_sales_missing,
 COUNT(CASE WHEN RtnMRP IS NULL THEN 1 END) AS rtnmrp_missing,
 COUNT(CASE WHEN TRIM(Formulation) = '' OR Formulation IS NULL
THEN 1 END) AS
formulation_missing,
 COUNT(CASE WHEN TRIM(DrugName) = '' OR DrugName IS NULL THEN
1 END) AS
drugname_missing,
 COUNT(CASE WHEN TRIM(SubCat) = '' OR SubCat IS NULL THEN 1 END)
AS subcat_missing,
 COUNT(CASE WHEN TRIM(SubCat1) = '' OR SubCat1 IS NULL THEN 1
END) AS subcat1_missing,
 COUNT(*) AS total_rows
FROM clean_data;
```



typeofsales_missing	patient_id_missing	specialisation_missing	dept_missing	dateofbill_missing	quantity_missing	returnquantity_missing	final_cost_
0	0	0	0	0	0	0	0



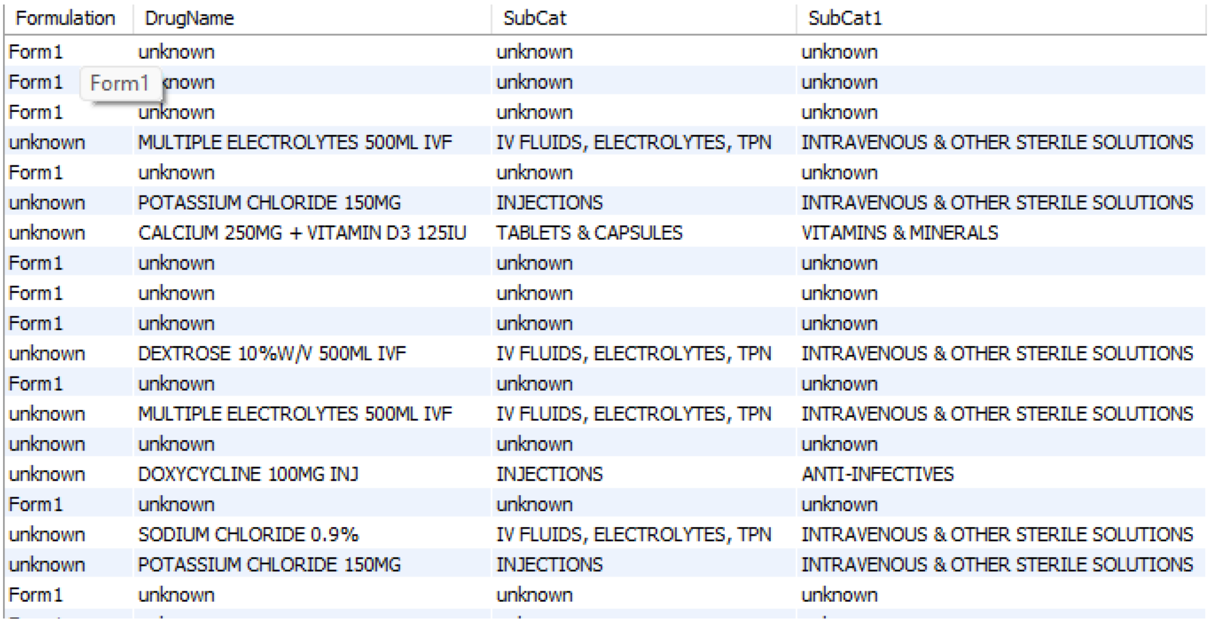
final_cost_missing	final_sales_missing	rtnmrp_missing	formulation_missing	drugname_missing	subcat_missing	subcat1_missing
0	0	0	653	1668	1668	1692

From the above result formulation, Drugname, SubCat, SubCat1 had missing values

8. Replacing missing values with unknown

```
UPDATE clean_data
SET
 Formulation = CASE WHEN Formulation = '' THEN 'unknown' ELSE
Formulation END;
UPDATE clean_data
SET
 DrugName = CASE WHEN DrugName = '' THEN 'unknown' ELSE
DrugName END;
UPDATE clean_data
SET
 SubCat = CASE WHEN SubCat = '' THEN 'unknown' ELSE SubCat END;
UPDATE clean_data
SET
 SubCat1 = CASE WHEN SubCat1 = '' THEN 'unknown' ELSE SubCat1 END;

select  Formulation, DrugName, SubCat, SubCat1 From clean_data
 where Formulation = 'unknown' or DrugName = 'unknown' or SubCat =
'unknown' or SubCat1 = 'unknown';
```



Formulation	DrugName	SubCat	SubCat1
Form1	unknown	unknown	unknown
Form1	unknown	unknown	unknown
Form1	unknown	unknown	unknown
unknown	MULTIPLE ELECTROLYTES 500ML IVF	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTIONS
Form1	unknown	unknown	unknown
unknown	POTASSIUM CHLORIDE 150MG	INJECTIONS	INTRAVENOUS & OTHER STERILE SOLUTIONS
unknown	CALCIUM 250MG + VITAMIN D3 125IU	TABLETS & CAPSULES	VITAMINS & MINERALS
Form1	unknown	unknown	unknown
Form1	unknown	unknown	unknown
Form1	unknown	unknown	unknown
unknown	DEXTROSE 10%W/V 500ML IVF	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTIONS
Form1	unknown	unknown	unknown
unknown	MULTIPLE ELECTROLYTES 500ML IVF	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTIONS
unknown	unknown	unknown	unknown
unknown	DOXYCYCLINE 100MG INJ	INJECTIONS	ANTI-INFECTIVES
Form1	unknown	unknown	unknown
unknown	SODIUM CHLORIDE 0.9%	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTIONS
unknown	POTASSIUM CHLORIDE 150MG	INJECTIONS	INTRAVENOUS & OTHER STERILE SOLUTIONS
Form1	unknown	unknown	unknown

9. Identifying and cleaning duplicate values based on Patient_ID.

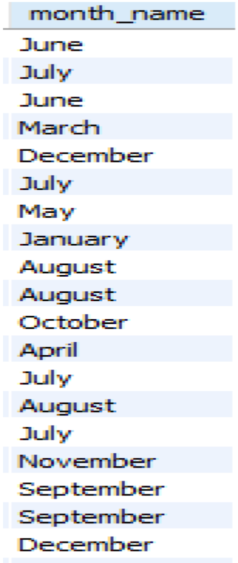
```
DELETE FROM clean_data
WHERE (Patient_ID, Dateofbill, DrugName) IN (
 SELECT t.Patient_ID, t.Dateofbill, t.DrugName
 FROM (
 SELECT Patient_ID, Dateofbill, DrugName
 FROM clean_data
 GROUP BY Patient_ID, Dateofbill, DrugName
 HAVING COUNT(*) > 1
 ) AS t
);
```

There are 125 duplicates in Patient_ID and we removed them.

10.Creating a column which has month names

```
Alter table clean_data
Add month_name Text;

update clean_data
set month_name = monthname(Dateofbill);
```

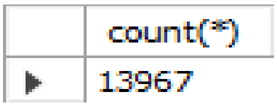


month_name
June
July
June
March
December
July
May
January
August
August
October
April
July
August
July
November
September
September
December

** EDA (Exploratory Data Analysis)

1. Cheking the total count of data and count of sales and returns.

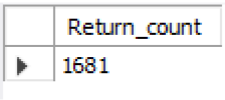
```
Select count(*) from clean_data;
```



	count(*)
▶	13967

Total data has 13967 rows.

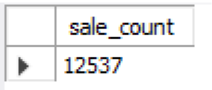
```
select count(*) as Return_count from medical_data where Typeofsales =
'Return';
```



	Return_count
▶	1681

Total data has 1681 rows where the medicines are returned.

```
select count(*) as sale_count from medical_data where Typeofsales =
'Sale';
```

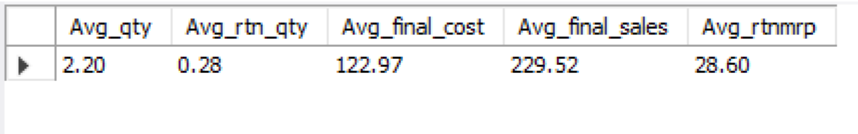


	sale_count
▶	12537

Total data has 12537 rows where the medicines are sold.

2. Mean

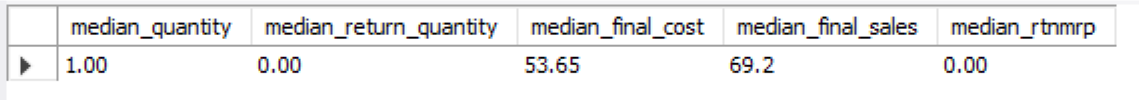
```
select
round(Avg(Quantity), 2) as Avg_qty,
round(Avg(ReturnQuantity), 2) as Avg_rtn_qty,
round(Avg(Final_Cost), 2) as Avg_final_cost,
round(Avg(Final_Sales), 2) as Avg_final_sales,
round(Avg(RtnMRP), 2) as Avg_rtnmrp from clean_data;
```



	Avg_qty	Avg_rtn_qty	Avg_final_cost	Avg_final_sales	Avg_rtnmrp
▶	2.20	0.28	122.97	229.52	28.60

3. Median

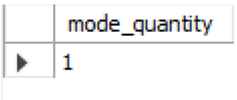
```
SELECT
 ROUND(AVG(Quantity), 2) AS median_quantity,
 ROUND(AVG(ReturnQuantity), 2) AS median_return_quantity,
 ROUND(AVG(Final_Cost), 2) AS median_final_cost,
 ROUND(AVG(Final_Sales), 2) AS median_final_sales,
 ROUND(AVG(RtnMRP), 2) AS median_rtnmrp
FROM (
 SELECT Final_Cost, Final_Sales, Quantity, ReturnQuantity, RtnMRP,
 ROW_NUMBER() OVER (ORDER BY Final_Cost) AS row_num,
 COUNT(*) OVER () AS total_rows FROM clean_data
) AS subquery
WHERE row_num IN (FLOOR((total_rows + 1) / 2), CEILING((total_rows +
1) / 2));
```



	median_quantity	median_return_quantity	median_final_cost	median_final_sales	median_rtnmrp
▶	1.00	0.00	53.65	69.2	0.00

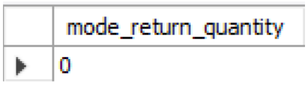
4. Mode

```
select Quantity as mode_quantity from clean_data group by Quantity
order by Count(*) desc limit 1;
```



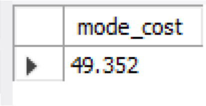
	mode_quantity
▶	1

```
select ReturnQuantity as mode_return_quantity from clean_data group
by ReturnQuantity order by Count(*) desc limit 1;
```



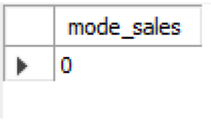
	mode_return_quantity
▶	0

```
select Final_Cost as mode_cost from clean_data group by Final_Cost
order by Count(*) desc limit 1;
```



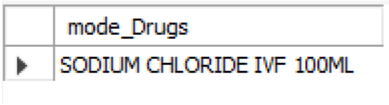
	mode_cost
▶	49.352

```
select Final_Sales as mode_sales from clean_data group by Final_Sales
order by Count(*) desc limit 1;
```



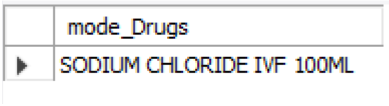
mode_sales
0

```
select DrugName as mode_Drugs from clean_data where DrugName <>
'unknown'
group by DrugName order by Count(*) desc limit 1;
```



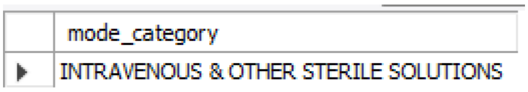
mode_Drugs
SODIUM CHLORIDE IVF 100ML

```
select SubCat as mode_category from clean_data group by SubCat order
by Count(*) desc limit 1;
```



mode_Drugs
SODIUM CHLORIDE IVF 100ML

```
select SubCat1 as mode_category from clean_data group by SubCat1
order by Count(*) desc limit 1;
```

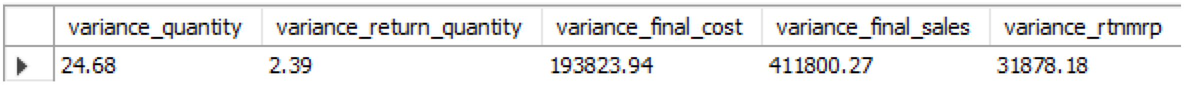


mode_category
INTRAVENOUS & OTHER STERILE SOLUTIONS

Observation: By the above results of mean, median and mode mean is greater than median and median is greater than mode i.e., mean>median>mode. So that the data is positively skewed.

5. Variance

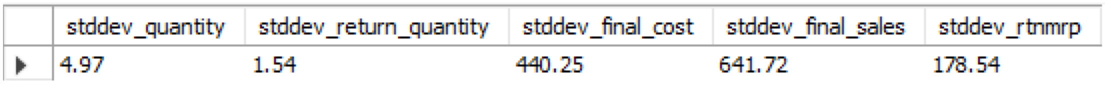
```
SELECT
 ROUND(VARIANCE(Quantity), 2) AS variance_quantity,
 ROUND(VARIANCE(ReturnQuantity), 2) AS variance_return_quantity,
 ROUND(VARIANCE(Final_Cost), 2) AS variance_final_cost,
 ROUND(VARIANCE(Final_Sales), 2) AS variance_final_sales,
 ROUND(VARIANCE(RtnMRP), 2) AS variance_rtnmrp
FROM clean_data;
```



	variance_quantity	variance_return_quantity	variance_final_cost	variance_final_sales	variance_rtnmrp
▶	24.68	2.39	193823.94	411800.27	31878.18

6. Standard Deviation

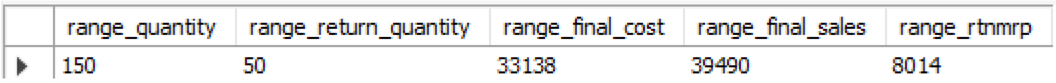
```
SELECT
 ROUND(STDDEV(Quantity), 2) AS stddev_quantity,
 ROUND(STDDEV(ReturnQuantity), 2) AS stddev_return_quantity,
 ROUND(STDDEV(Final_Cost), 2) AS stddev_final_cost,
 ROUND(STDDEV(Final_Sales), 2) AS stddev_final_sales,
 ROUND(STDDEV(RtnMRP), 2) AS stddev_rtnmrp
FROM clean_data;
```



	stddev_quantity	stddev_return_quantity	stddev_final_cost	stddev_final_sales	stddev_rtnmrp
▶	4.97	1.54	440.25	641.72	178.54

7. Range

```
SELECT
 MAX(Quantity) - MIN(Quantity) AS range_quantity,
 MAX(ReturnQuantity) - MIN(ReturnQuantity) AS range_return_quantity,
 MAX(Final_Cost) - MIN(Final_Cost) AS range_final_cost,
 MAX(Final_Sales) - MIN(Final_Sales) AS range_final_sales,
 MAX(RtnMRP) - MIN(RtnMRP) AS range_rtnmrp
FROM clean_data;
```

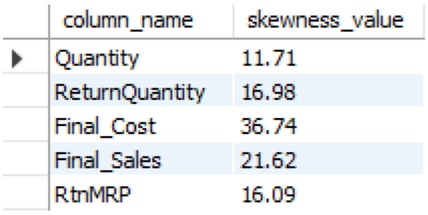


	range_quantity	range_return_quantity	range_final_cost	range_final_sales	range_rtnmrp
▶	150	50	33138	39490	8014

Observation: From the results of measures of dispersion i.e., variance, standard deviation, range. The values are too high in the columns final_cost, final_sales, rtnmrp. The values are too dispursed in these columns.

8. Skewness

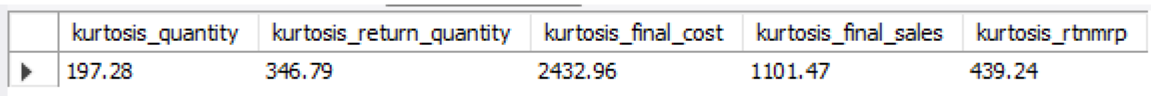
```
SELECT 'Quantity' AS column_name,
ROUND((SUM(POW(Quantity - (SELECT AVG(Quantity) FROM
clean_data), 3)) / (COUNT(*) *
POW(STDDEV(Quantity), 3))), 2) AS skewness_value
FROM clean_data
UNION ALL
SELECT 'ReturnQuantity' AS column_name,
ROUND((SUM(POW(ReturnQuantity - (SELECT AVG(ReturnQuantity)
FROM clean_data), 3)) /
(COUNT(*) * POW(STDDEV(ReturnQuantity), 3))), 2) AS skewness_value
FROM clean_data
UNION ALL
SELECT 'Final_Cost' AS column_name,
ROUND((SUM(POW(Final_Cost - (SELECT AVG(Final_Cost) FROM
clean_data), 3)) / (COUNT(*) *
POW(STDDEV(Final_Cost), 3))), 2) AS skewness_value
FROM clean_data
UNION ALL
SELECT 'Final_Sales' AS column_name,
ROUND((SUM(POW(Final_Sales - (SELECT AVG(Final_Sales) FROM
clean_data), 3)) / (COUNT(*) *
POW(STDDEV(Final_Sales), 3))), 2) AS skewness_value
FROM clean_data
UNION ALL
SELECT 'RtnMRP' AS column_name,
ROUND((SUM(POW(RtnMRP - (SELECT AVG(RtnMRP) FROM clean_data),
3)) / (COUNT(*) *
POW(STDDEV(RtnMRP), 3))), 2) AS skewness_value
FROM clean_data;
```



column_name	skewness_value
Quantity	11.71
ReturnQuantity	16.98
Final_Cost	36.74
Final_Sales	21.62
RtnMRP	16.09

9. Kurtosis

```
SELECT
ROUND((SUM(POWER(Quantity - avg_value, 4)) / (COUNT(Quantity) *
POWER(STDDEV(Quantity),
4))), 2) AS kurtosis_quantity,
ROUND((SUM(POWER(ReturnQuantity - avg_value, 4)) /
(COUNT(ReturnQuantity) *
POWER(STDDEV(ReturnQuantity), 4))), 2) AS kurtosis_return_quantity,
ROUND((SUM(POWER(Final_Cost - avg_value, 4)) / (COUNT(Final_Cost) *
POWER(STDDEV(Final_Cost), 4))), 2) AS kurtosis_final_cost,
ROUND((SUM(POWER(Final_Sales - avg_value, 4)) / (COUNT(Final_Sales)
*
POWER(STDDEV(Final_Sales), 4))), 2) AS kurtosis_final_sales,
ROUND((SUM(POWER(RtnMRP - avg_value, 4)) / (COUNT(RtnMRP) *
POWER(STDDEV(RtnMRP), 4))),
2) AS kurtosis_rtnmrp
FROM
 (SELECT
 AVG(Quantity) AS avg_value,
 STDDEV(Quantity) AS stddev_value,
 COUNT(Quantity) AS count_value
 FROM clean_data) AS subquery, clean_data;
```

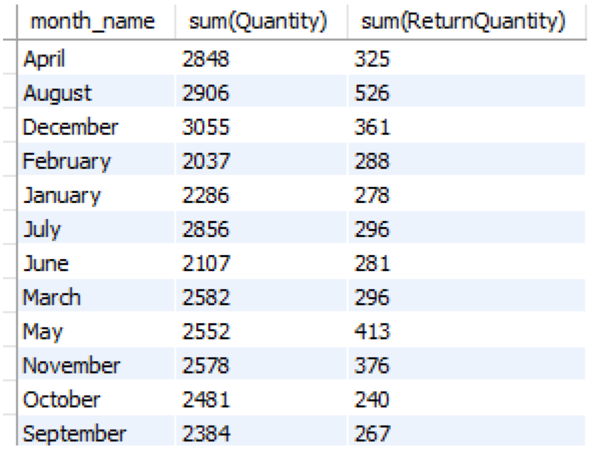


	kurtosis_quantity	kurtosis_return_quantity	kurtosis_final_cost	kurtosis_final_sales	kurtosis_rtnmrp
▶	197.28	346.79	2432.96	1101.47	439.24

** BUSINESS INSIGHTS

- Finding the most effective month(more sales) and most ineffective month (more returns).

```
select month_name, sum(Quantity), sum(ReturnQuantity)
from clean_data
group by month_name
order by month_name;
```

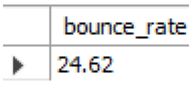


month_name	sum(Quantity)	sum(ReturnQuantity)
April	2848	325
August	2906	526
December	3055	361
February	2037	288
January	2286	278
July	2856	296
June	2107	281
March	2582	296
May	2552	413
November	2578	376
October	2481	240
September	2384	267

Observation : Maximum quantity sold in December month and least quantity sold in February. Maximum returns are in the august month and least return in the October month

- Bounce rate

```
SELECT ROUND((bounced_customers / total_customers) * 100, 2) AS
bounce_rate
FROM
 (SELECT COUNT(DISTINCT Patient_ID) AS total_customers
 FROM clean_data
 WHERE Typeofsales IN ('Sale', 'Return')) AS t1,
 (SELECT COUNT(DISTINCT Patient_ID) AS bounced_customers
 FROM clean_data
 WHERE Typeofsales = 'Return' AND Final_Sales = 0) AS t2;
```

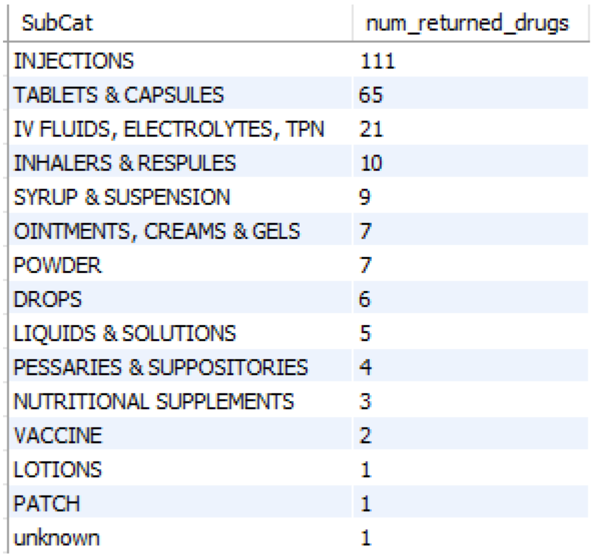


bounce_rate
24.62

Observation : There is a bounce rate of 24.62% that means for every 100 customers ther are approximately 25 customers who are returning the medicine. This is a significance percentage which causes dissatisfaction to the customer.

- Checking Category wise medicine returns

```
SELECT SubCat, COUNT(DISTINCT DrugName) AS num_returned_drugs
FROM clean_data
WHERE Typeofsales = 'Return' AND Final_Sales = 0
GROUP BY SubCat
ORDER BY num_returned_drugs DESC;
```

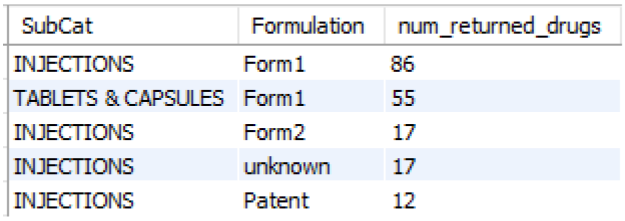


SubCat	num_returned_drugs
INJECTIONS	111
TABLETS & CAPSULES	65
IV FLUIDS, ELECTROLYTES, TPN	21
INHALERS & RESPULES	10
SYRUP & SUSPENSION	9
OINTMENTS, CREAMS & GELS	7
POWDER	7
DROPS	6
LIQUIDS & SOLUTIONS	5
PESSARIES & SUPPOSITORIES	4
NUTRITIONAL SUPPLEMENTS	3
VACCINE	2
LOTIONS	1
PATCH	1
unknown	1

- Finding which formulation has highest returns

```
SELECT SubCat, Formulation, COUNT(DISTINCT DrugName) AS
num_returned_drugs
FROM clean_data
WHERE Typeofsales = 'Return' AND Final_Sales = 0
```

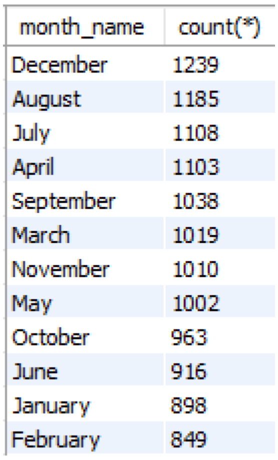
```
GROUP BY SubCat, Formulation
ORDER BY num_returned_drugs DESC limit 5;
```



SubCat	Formulation	num_returned_drugs
INJECTIONS	Form1	86
TABLETS & CAPSULES	Form1	55
INJECTIONS	Form2	17
INJECTIONS	unknown	17
INJECTIONS	Patent	12

- Finding which months has highest and lowest medicine returns

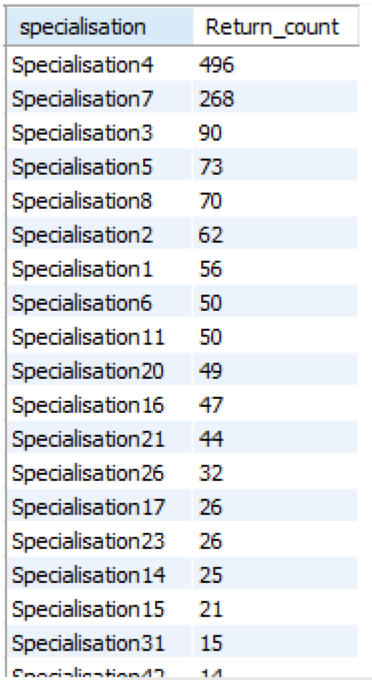
```
select month_name, count(*) from clean_data where Typeofsales =
'Sale' group by month_name;
```



month_name	count(*)
December	1239
August	1185
July	1108
April	1103
September	1038
March	1019
November	1010
May	1002
October	963
June	916
January	898
February	849

- Checking which specialisation has highest medicine returns

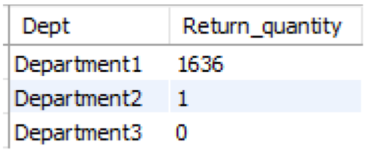
```
select specialisation,
sum(case when Typeofsales = 'Return' then 1 else 0 end) as
'Return_count'
from clean_data
group by specialisation;
```



specialisation	Return_count
Specialisation4	496
Specialisation7	268
Specialisation3	90
Specialisation5	73
Specialisation8	70
Specialisation2	62
Specialisation1	56
Specialisation6	50
Specialisation11	50
Specialisation20	49
Specialisation16	47
Specialisation21	44
Specialisation26	32
Specialisation17	26
Specialisation23	26
Specialisation14	25
Specialisation15	21
Specialisation31	15
Specialisation42	14

- Checking which department has highest medicine returns

```
select Dept,
sum(case when Typeofsales = 'Return' then 1 else 0 end) as
'Return_quantity'
from clean_data
group by Dept
order by Dept;
```



Dept	Return_quantity
Department1	1636
Department2	1
Department3	0

Conclusion :

- Based on subcategory injections, Tablets and capsules has higher returns so there is need of special attention on this categories that may reduce the returns and bounce rate.

- Formulation 1 has higher returns on both injections and tablets.

- If we categorize the return count based on months we can see that there are higher returns in the month of may.

- There need to be efficient inventory management in the hospital and pharmacy which can be done by good communication between the hospitals and suppliers which reduces bounce rate.